UML Modeling Business Rules for caCORE 3.2

Contents

Introduction

This document describes the UML Modeling syntax required by the caCORE 3.2 Product suite. This includes all product iterative releases prefixed with "3.2". Any modeling syntax not explicitly noted is ignored by the caCORE Products. The caCORE product business rules specific to each UML entity are detailed below.

Refer also to UML Modeling FAQs and XMI Tag Reference.

Audience

This document is intended for the following audiences:

1. UML Modelers for an understanding how the UML syntax is interpreted.
2. Developers and caCORE Training as an appendix to the caCORE SDK Programmers Guide.
3. caCORE QA as a set of business rules to verify the product features.
4. caCORE Development Teams to ensure consistency across all products.

Note: some prior knowledge of the caCORE Products is required.

Organization

This document is organized as a reference to the UML Modeling syntax with details for each product's interpretation of the syntax. It is not a complete UML Modeling description. Only the syntax required and referenced by the caCORE Products is included, e.g. no reference to a Class Method exists because all caCORE Products ignore them. Please see other reference sources for a complete description of UML.

Each caCORE Product is noted separately with each UML construct to avoid any ambiguity or question of collateral effects. When a product does not appear with a Construct or Property it indicates the development team has confirmed it is not applicable and is consequently ignored. The appearance of "UNKNOWN" indicates the construct has not yet been reviewed with the product development team.

UML Models Supported by caCORE

A UML Class Diagram is the only kind of UML model that the caCORE process currently supports. Class diagrams are the most common UML

diagram. They represent the static items that exist in a system, their structure, and their relationships. Class diagrams are usually used to depict the logical (e.g. logical model) and physical (e.g. data model) design of the system.

Definitions & Acronyms

- **caAdapter**
  Tool for mapping the object model to data model
- **caDSR**
  NCI Cancer Data Standards Repository - stores models in database (ISO 11179 Metadata Registry)
- **caDSR Grid Service**
  Provides access to UML view of caDSR models (uses UML model browser's model); creates caGrid metadata
- **caGrid Data Services / CQL**
  Provides access to data by querying using a UML-inspired query language
- **CLM**
  Common Logging Module - provides capability for Object and attribute level logging
- **CSM**
  Common Security Module - provides capability for Object and attribute level secruity
- **diagram**
  A plan, sketch, drawing or outline designed to demonstrate or explain how something works or to clarify the relationship between the parts of a whole. Example: UML Class Diagram.
- **SC**
  Semantic Connector - parses XML elements and searches EVS, annotates the UML Model using caCORE specific tags
- **SDK**
  caCORE SDK Code Generator - generates APIs, web services, Happy.jsp
- **SIW**
  Semantic Integration Workbench - reads the exported UML Model file and provides a UI for matching model elements with EVS and caDSR content; annotates the Model using caCORE specific tags
- **UML Loader**
  Transfoms the exported UML Model representation of XML Elements into caDSR
- **UML Model Browser**
  Provides viewing of the models loaded in caDSR
- **XML Utility**
  Generates xsd for UML Model from caDSR metadata

UML Entities

A brief quick reference UML Compatibility Matrix is also available.

Class

definition: a class is a representation of a generalized thing such as an animaal, a person, a car, a ticket or a system (Roff, 2003)

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- All classes should appear in the Logical Model package.

SIW

- May be stereotyped with "CADSR Value Domain" or "enumeration" to indicate that this class represents a Value Domain.
- Required Tags
    - CADSR_Description: contains a value describing the class (or value domain, if stererotyped) purpose. The value for this tag must not exceed 255 characters. If the class purpose exceeds 255 characters, extra "CADSR_DescriptionN" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - NOTE: The deprecated tag "documentation", along with "documentationN" can still be used in place of the "CADSR_Description" tags to accomodate models annotated prior to SIW 3.2.1.4.

UML Loader

- If the UML class is not stereotyped as "CADSR Value Domain" or "enumeration", then the UML class will be loaded into caDSR as an Object Class.
- If the UML class is stereotyped as "CADSR Value Domain" or "enumeration", then the UML class will be loaded into caDSR as a Value Domain.
- The UML class and attribute together are stored in the caDSR as a DEC.
- The UML class and attribute and data type (via the Value Domain) together are stored in the caDSR as a CDE.
- Required Tags for UML Class not stereotyped as "CADSR Value Domain" or "enumeration"
    - CADSR_Description, or the deprecated "documentation" tags, as defined for the SIW above.
    - Tags that map concepts to the class UNLESS at least one attribute in the class is mapped to a CDE, in which case the concept mapping is defined by the existing Object Class in that CDE.
        - ObjectClassConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this class.
        - ObjectClassConceptPreferredName: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this class.
        - ObjectClassConceptPreferredDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this class. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "ObjectClassConceptPreferredDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - ObjectClassConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this class.
        - ObjectClassConceptQualifierCodeN, where N is the qualifier position: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this class
        - ObjectClassConceptQualifierPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this class.
        - ObjectClassConceptQualifierPreferredDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this class. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "ObjectClassConceptPreferredDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - ObjectClassConceptQualifierDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this class.
- Required Tags for UML Class stereotyped as "CADSR Value Domain" or "enumeration"
    - CADSR_ValueDomainDefinition: contains a definition of the Value Domain
    - CADSR_ValueDomainDatatype: contains the type of data contained in the Value Domain. The data type must be one of the data types currently registered in caDSR.
    - CADSR_ValueDomainType: contains the type of Value Domain; "E" (enumerated) when the Value Domain contains a list of values, or "N" (non-enumerated) when the Value Domain does not contain a list of values.
    - CADSR_ConceptualDomainPublicID: contains the caDSR public id of the conceptual domain under which this Value Domain will be registered.
    - CADSR_ConceptualDomainVersion: contains the caDSR version of the conceptual domain under which this Value Domain will be registered.
    - CADSR_RepresentationPublicID: contains the caDSR public id of the representation term describing the data held in this Value Domain.
    - CADSR_RepresentationVersion: contains the caDSR version of the representation term describing the data held in this Value Domain.
- The final description for the class will be the concatenation of all the "CADSR_Description" tags, not to exceed 2000 characters in total. This description will be loaded into caDSR as an Alternate Description for the Object Class associated with the mapped CDE.
- NOTE: "CADSR_Description" tags are currently ignored for UML classes that are stereotyped as "CADSR Value Domain" or "enumeration".

XML Utility

- UNKNOWN

Class Name

definition: the text label for a UML Class

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- Must not be Java reserved words, e.g. int, long, void, etc.

SIW

- If camel cased, the name is broken apart by semantic connector using the capitalization to search for matching concepts in the EVS.

UML Loader

- If the class is not stereotyped as "enumeration" or "CADSR Value Domain":
    - The UML class name is used to create an alternate name for the Object Class that represents the UML class in caDSR.
    - The package name(s) and UML class name are concatenated to create a fully qualified alternate name for the Object Class that represents the UML class in caDSR.
    - The UML class and attribute names are concatenated to create an alternate name for the Data Element Concept (DEC) that represents the UML class attribute in caDSR.
    - The UML class and attribute names are concatenated to create an alternate name for the Data Element (CDE) that represents the UML class attribute in caDSR.
    - The package name(s) and the UML class and attribute names are concatenated to create a fully qualified alternate name for the Data Element (CDE) that represents the UML class attribute in caDSR.
- If the class is stereotyped as "enumeration" or "CADSR Value Domain":
    - The UML class name is used to create the Long Name for the Value Domain that represents the stereotyped class in caDSR.

Class Attribute

definition: characteristic or property of a class

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- Required Tags
    - CADSR_Description: contains a value describing the attribute purpose. The value for this tag must not exceed 255 characters. If the class purpose exceeds 255 characters, extra "CADSR_DescriptionN" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - NOTE: The deprecated tag "description", along with "descriptionN" can still be used in place of the "CADSR_Description" tags to accomodate models annotated prior to SIW 3.2.1.4.
    - CADSR Local Value Domain: only required IF the attribute is to use a Local Value Domain to represent its data type in caDSR. The value of this tag must be the name of a UML Class in the Logical Model that is stereotyped as "CADSR Value Domain" or "enumeration".

UML Loader

- If the UML class containing this attribute is not stereotyped as "CADSR Value Domain" or "enumeration", then the UML attribute will be loaded into caDSR as a Property.
    - The UML class and attribute together are stored in the caDSR as a DEC.
    - The UML class and attribute and data type (via the Value Domain) together are stored in the caDSR as a CDE.
- If the UML class containing this attribute is not stereotyped as "CADSR Value Domain" or "enumeration", then the definition in the CADSR_Description tags of the UML attribute provides an Alternative Definition for the CDE created for this class attribute.

- If the UML class containing this attribute is stereotyped as "CADSR Value Domain" or "enumeration", then the UML attribute will be loaded into caDSR as the Permissible Value of the Value Domain represented by this stereotyped class.
- Required Tags for UML Attribute in a UML Class not stereotyped as "CADSR Value Domain" or "enumeration"
    - CADSR_Description, or the deprecated "description" tags, as defined for the SIW above.
    - Tags required to map concepts to the attribute.
        - PropertyConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this attribute.
        - PropertyConceptPreferredName: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this attribute.
        - PropertyConceptDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this attribute. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "PropertyConceptDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - PropertyConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this attribute.
        - PropertyQualifierConceptCodeN, where N is the qualifier position: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this attribute.
        - PropertyQualifierConceptPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this attribute.
        - PropertyQualifierConceptDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this attribute. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "PropertyQualifierConceptDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - PropertyQualifierDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this attribute.
    - Tags required to map CDEs to the attribute.
        - CADSR_DE_ID: contains the public id of the existing CDE in caDSR that is mapped to the UML class attribute.
        - CADSR_DE_VERSION: contains the version number of the existing CDE in caDSR that is mapped to the UML class attribute.
    - Tags required to map an existing Value Domain to the attribute.
        - CADSR_VD_ID: contains the public id of the existing Value Domain in caDSR that is mapped to UML class attribute.
        - CADSR_VD_VERSION: contains the version number of the existing Value Domain in caDSR that is mapped to the UML class attribute.
    - Tags required to map Local Value Domains to the attribute.
        - CADSR Local Value Domain: contains the name of the Local Value Domain (represented in the UML model as a UML Class stereotyped as "CADSR Value Domain" or "enumeration").
    - NOTE: A UML attribute can contain tags for any combination of concept mapping, CDE mapping, existing Value Domain mapping, or Local Value Domain mapping. However, the SIW and UML Loader will only recognize the tags using the following precedence
        - CDE mapping overrides concept mapping
        - CDE mapping overrides existing Value Domain mapping or Local Value Domain mapping
        - existing Value Domain mapping overrides Local Value Domain mapping
- Required Tags for UML attribute in a UML Class stereotyped as "CADSR Value Domain" or "enumeration"
    - CADSR_Description, or the deprecated "description" tags, as defined for the SIW above.
    - Tags required to map concepts to the permissible value (the UML attribute). NOTE: concept mapping to permissible values is optional.
        - ValueMeaningConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this permissible value.
        - ValueMeaningConceptPreferredName: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this permissible value.
        - ValueMeaningConceptDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this permissible value. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "ValueMeaningConceptDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - ValueMeaningConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this permissible value.
        - ValueMeaningQualifierConceptCodeN, where N is the qualifier position: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this permissible value.
        - ValueMeaningQualifierConceptPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this permissible value.
        - ValueMeaningQualifierConceptDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this permissible value. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "ValueMeaningQualifierConceptDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
        - ValueMeaningQualifierDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this permissible value.
- For attributes of a class not stereotyped as "enumeration" or "CADSR Value Domain":
    - The attribute is stored in the caDSR as a Property.
    - The class and attribute together are stored in the caDSR as a DEC.
    - The class and attribute and data type (via the Value Domain) together are stored in the caDSR as a CDE.
- For attributes of a class stereotyped as "enumeration" or "CADSR Value Domain", the attribute is stored in caDSR as a Permissible Value/Value Meaning pair.

- The Permissible Value will be created from the name of the attribute.
- The Value Meaning created in the caDSR will be created from:
    - If there are no concepts mapped to the attribute representing the Permissible Value:
        - The Value Meaning Name will be created from the name of the attribute.
        - The Value Meaning Concept Codes will be left blank.
        - The Value Meaning Description will be created from the concatenation of all the "CADSR_Description" tags, not to exceed 2000 characters in total.
    - If there are concepts mapped to the attribute representing the Permissible Value:
        - The Value Meaning Name will be created from the concatenation of the names of the qualifier and primary concepts mapped to the attribute.
        - The Value Meaning Concept Codes will be created from the concatenation of the codes of the qualifier and primary concepts mapped to the attribute.
        - The Value Meaning Description will be created from the concatenation of the definitions of the qualifier and primary concepts mapped to the attribute.

Class Attribute Name

definition: text label for a class attribute

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- Must not be Java reserved words, e.g. int, long, void, etc.

SIW

- If camel cased, the name is broken apart by semantic connector, using the capitalization to search for matching concepts in the EVS.

UML Loader

- If the attribute belongs to a UML class that is stereotyped as "enumeration" or "CADSR Value Domain":
    - The UML attribute name is used to create the Permissible Value in caDSR.
- If the attribute belongs to a UML class that is not stereotyped as "enumeration" or "CADSR Value Domain":
    - The UML attribute name is used to create the alternate name of the Property that represents the UML attribute in caDSR.
    - The UML class and attribute names are concatenated to create an alternate name for the Data Element Concept (DEC) that represents the UML class attribute in caDSR.
    - The UML class and attribute names are concatenated to create an alternate name for the Data Element (CDE) that represents the UML class attribute in caDSR.
    - The package name(s) and the UML class and attribute names are concatenated to create a fully qualified alternate name for the Data Element (CDE) that represents the UML class attribute in caDSR.

Class Attribute Data Type

definition: describes the kinds of values that can be assigned to the attribute

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- The data type must be one of the following Java data types:
    - Java wrapper class

- Java Collection<type> where type is one of:
  - int
  - integer
  - double
  - float
  - long
  - string
  - boolean
  - byte
  - short
  - char
  - character

SIW

- Must be recognized as valid data type according to datatype-mapping.xml
- If the attribute containing any other data type is mapped to an existing CDE, an existing Value Domain or a Local Value Domain, then the UML attribute's data type is ignored by the SIW and the mapped Value Domain is used instead.

UML Loader

- Must be recognized as valid data type according to datatype-mapping.xml
- Stored in the caDSR as the Value Domain that maps to the data type in datatype-mapping.xml
- If the attribute containing any other data type is mapped to an existing CDE, an existing Value Domain or a Local Value Domain, then the UML attribute's data type is ignored by the UML Loader and the mapped Value Domain is used instead.

Abstract Class

definition: a general class that is created to supress unnecessary details and instead focus on a specific set of details that are of interest to the system being modeled

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- Treated as a Class, the abstract quality is lost.

UML Loader

- Treated as a Class, the abstract quality is lost.

Inherited Class Attributes

definition:

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- Select the Display Inherited Attributes preference in SIW to see inherited attributes listed under each child class.
- Will derive a definition by concatenating the definitions from the CADSR_Description tags in the child class and the parent attribute
- Can map an inherited attribute to a CDE or a Value Domain or Local Value Value Domain using the SIW.
- If an inherited attribute is mapped to a Value Domain, you cannot map it to a CDE.
- If an inherited attribute is mapped to a CDE, you cannot map it to a Value Domain.
- Required Tags
    - CADSR_Inherited.<attributeName>Local Value Domain : only required IF the inherited attribute is to use a Local Value Domain to represent its data type in caDSR. <attributeName> should be the name of the inherited attribute. The value of this tag must be the name of a UML Class in the Logical Model that is stereotyped as "CADSR Value Domain" or "enumeration". This tag is placed on the CHILD CLASS containing the inherited attribute.

UML Loader

- Will create one CDE for each inherited attribute in a class.
- If the inherited attribute in the child class contains no tags, then:
    - A DEC will be created using the Object Class of the child class and the Property of the parent attribute.
    - A CDE will be created using that DEC and the Value Domain of the parent attribute.
- If the inherited attribute in the child class is mapped to a Value Domain or a Local Value Domain, then:
    - A DEC will be created using the Object Class of the child class and the Property of the parent attribute.
    - A CDE will be created using that DEC and the Value Domain mapped to the child attribute.
    - The Value Domain of the parent attribute will be ignored.
- If the inherited attribute in the child class is mapped to a CDE, then:
    - The mapped CDE will be reused for this inherited attribute
    - The Object Class of the child class and the Property and Value Domain of the parent attribute will be ignored.
- The definition in the CADSR_Description tags from the child class is concatenated with the definition in the CADSR_Description tags of the parent attribute to provide an Alternative Definition for the CDE created for this inherited attribute.
- Required Tags
    - Tags required to map CDEs to the inherited attribute. NOTE: These tags are added to the child CLASS containing the iniherited attribute.
        - CADSR_Inherited.<attributeName>.DE_ID: contains the public id of the existing CDE in caDSR that is mapped to the inherited attribute. <attributeName> should be the name of the inherited attribute.
        - CADSR_Inherited.<attributeName>.DE_VERSION: contains the version number of the existing CDE in caDSR that is mapped to the inherited attribute. <attributeName> should be the name of the inherited attribute.
    - Tags required to map an existing Value Domain to the inherited attribute. NOTE: These tags are added to the child CLASS containing the iniherited attribute.
        - CADSR_Inherited.<attributeName>.VD_ID: contains the public id of the existing Value Domain in caDSR that is mapped to the inherited attribute. <attributeName> should be the name of the inherited attribute.
        - CADSR_Inherited.
          Unknown macro: {1}

          .VD_VERSION: contains the version number of the existing Value Domain in caDSR that is mapped to the inherited attribute. <attributeName> should be the name of the inherited attribute.
    - Tags required to map Local Value Domains to the attribute. NOTE: This tag is added to the child CLASS containing the iniherited attribute.
        - CADSR_Inherited.<attributeName>.Local Value Domain: contains the name of the Local Value Domain (represented in the UML model as a UML Class stereotyped as "CADSR Value Domain" or "enumeration"). <attributeName> should be the name of the inherited attribute.
    - NOTE: An inherited attribute can contain tags for any combination of CDE mapping, existing Value Domain mapping, or Local Value Domain mapping. However, the SIW and UML Loader will only recognize the tags using the following precedence:
        - CDE mapping overrides the default mapping of child Object Class, parent attribute Property and parent attribute Value Domain.
        - CDE mapping overrides existing Value Domain mapping or Local Value Domain mapping.
        - Existing Value Domain mapping overrides Local Value Domain mapping.
- The final description for the inherited attribute will be the concatenation of the "CADSR_Description" tags, not to exceed 2000 characters in total, from the child UML class and the parent UML attribute. This description will be loaded into caDSR as an Alternate Description for the CDE associated with the child class's inherited attribute.

Class Association

definition:

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- A regular association must contain:
    - directionality
    - name, minimally on each end of the association that contains an arrow
    - multiplicity on each end
- A generalization association represents inheritance between two classes. The arrow end of the generalization must point to the parent class.

UML Loader

- Stored as an Object Class Relationship
- Required tags when adding semantics to regular associations. NOTE: Adding semantics to associations is optional.
    - AssociationRoleConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's role.
    - AssociationRoleConceptPreferredNam: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's role.
    - AssociationRoleConceptDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's role. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationRoleConceptDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - AssociationRoleConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's role.
    - AssociationRoleQualifierConceptCodeN, where N is the qualifier position: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's role.
    - AssociationRoleQualifierConceptPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's role.
    - AssociationRoleQualifierConceptDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's role. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationRoleQualifierConceptDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - AssociationRoleQualifierConceptDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's role.
    - AssociationSourceConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's source.
    - AssociationSourceConceptPreferredNam: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's source.
    - AssociationSourceConceptDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's source. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationSourceConceptDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - AssociationSourceConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's source.
    - AssociationSourceQualifierConceptCodeN, where N is the qualifier position: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's source.
    - AssociationSourceQualifierConceptPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's source.
    - AssociationSourceQualifierConceptDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's source. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationSourceQualifierConceptDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - AssociationSourceQualifierConceptDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's source.
    - AssociationRoleConceptCode: contains the concept code specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's role.
    - AssociationTargetConceptPreferredNam: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's target.
    - AssociationTargetConceptDefinition: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's target. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationTargetConceptDefinition_N" tags may be added, where N is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
    - AssociationTargetConceptDefinitionSource: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the primary concept mapped to this association's target.
    - AssociationTargetQualifierConceptCodeN, where N is the qualifier position: contains the concept code specified in the ontology

(e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's target.
- AssociationTargetQualifierConceptPreferredNameN, where N is the qualifier position: contains the concept preferred name specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's target.
- AssociationTargetQualifierConceptDefinitionN, where N is the qualifier position: contains the concept preferred definition specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's target. The value for this tag must not exceed 255 characters. If the concept preferred definition exceeds 255 characters, extra "AssociationTargetQualifierConceptDefinitionN_n" tags may be added, where n is a number between 2 and 8. Again, each of these tag values must not exceed 255 characters.
- AssociationTargetQualifierConceptDefinitionSourceN, where N is the qualifier position: contains the concept definition source specified in the ontology (e.g., NCI Thesaurus) for the Nth qualifier concept mapped to this association's target.

Association Directionality

definition:

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- Must be Bi-Directional, Source->Destination or Destination->Source.

SIW

- Must be Bi-Directional, Source->Destination or Destination->Source.

UML Loader

- Stored as an attribute of Object Class Relationship (OCR).

Association Multiplicity

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- Must be specified on both sides of the association, regardless of the directionality.

SIW

- Must be specified on both sides of the association, regardless of the directionality.

UML Loader

- Stored as an attribute of Object Class Relationship (OCR).

Association Role Names

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- Must have a role name for at least each end of the assocation that has an arrow:
    - both sides for bi-directional
    - at least target side for Source -> Destination
    - at least source side for Destination -> Source

SIW

- Must have a role name for at least each end of the assocation that has an arrow:
    - both sides for bi-directional
    - at least target side for Source -> Destination
    - at least source side for Destination -> Source

UML Loader

- Stored as property of OCR

Generalization (Inheritance)

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- Can not have multiple inheritance (i.e., one class cannot directly inherit from more than one parent class)
- Any depth of inheritance is okay (i.e., Class A can inherit from Class B that inherits from Class C, etc.)

UML Loader

- Stored as OCR of type "IS_A"
- Can not have multiple inheritance (i.e., one class cannot directly inherit from more than one parent class)
- Any depth of inheritance is okay (i.e., Class A can inherit from Class B that inherits from Class C, etc.)

Aggregation

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- The quality is lost

UML Loader

- Stored as simple associations, the aggregation quality is lost.

Composition

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- The composition quality is lost.

UML Loader

- Stored as simple associations, the composition quality is lost.

Enumerated Values in a Stereotyped Class (Value Domain)

caAdapter

- UNKNOWN

caGrid

- UNKNOWN

CSM

- UNKNOWN

SDK

- UNKNOWN

SIW

- Refer to Class Attribute section.

UML Loader

- Refer to Class Attribute section.